IN THE SPECIFICATION

Please amend Page 3, Lines 3-10 to read as follows:

A1

In microprocessors employing pipelined architecture, it is desirable to be in the process of executing as many instructions as possible, so that each element of the pipeline is maintained busy. However, some instructions, such as instructions that load data from external memory or store data into external memory, must generally be executed in their original sequence order, so as to avoid the external memory ever being in an incorrect state. Moreover, when such instructions refer to identical external memory locations, there is no particular need to wait for the actual external memory operations to complete, as the identical data is already available for the processor to operate with.

Please amend Page 3, Line 18 to Page 4, Line 4 to read as follows:

A2

Accordingly, it would be advantageous to provide a technique for operating a pipelined microprocessor more quickly, by detecting instructions that load from identical memory locations as were recently stored to, without having to actually compute the referenced external memory addresses. In a preferred embodiment, the microprocessor examines the symbolic structure of instructions as they are encountered, so as to be able to detect identical memory locations by examination of their symbolic structure. For example, instructions that store to and load from an identical offset from an identical register are determined to be referencing the identical memory location, without having to actually compute the complete physical target address.

Please amend Page 6, Line 21 to Page 7, Line 3 to read as follows:

A3

The microprocessor 100 reads a sequence of instructions 151 from the instruction memory 150 using the instruction fetch stage 110 (and including any associated memory read or write elements in the microprocessor 100). In a preferred embodiment, the input instruction buffer 110 includes a plurality of instructions 151 from the instruction memory 150, but there is no particular requirement therefor.

Please amend Page 8, Lines 1 to 4, to read as follows:

A4

Similarly, an instruction that stores data to external memory has a format that refers to the specific location in external memory into which to store the data. The format can similarly include a base address value and an offset address value, which are used to compute the effective reference address of the instruction 151.

Please amend Page 10, Lines 1 to 5, to read as follows:

A5

Although the actual first (store) instruction 151 would be physically performed and completed by external memory, the microprocessor 100 can proceed without physically performing the second (load) instruction 151. Instead, the microprocessor 100 can use the identical data from its internal register, thus removing a relative delay in microprocessor 100 operation.

Please amend Page 12, Lines 8 to 15, to read as follows:

A6

At a step 217, the bypass element 121 determines whether the operand addresses that the instructions 151 refer to include identical base address values and offset address values. If so, the bypass element 121 generates a bypass signal indicating that the instructions 151 refer to the same location in external memory. If not, the bypass element 121 does not generate a bypass signal. (In alternative embodiments, the bypass element 121 may generate an inverse bypass signal). If the bypass element 121 generates a bypass signal, the method 200 proceeds with the step 220. If not, the method 200 proceeds with the step 221.